



# Decentralized Proportional Load Balancing

Jonatha Anselmi, N. S. Walton

## ► To cite this version:

Jonatha Anselmi, N. S. Walton. Decentralized Proportional Load Balancing. SIAM Journal on Applied Mathematics, 2016, 76 (1), pp.391-410. 10.1137/140969361 . hal-01415856

**HAL Id: hal-01415856**

**<https://hal.science/hal-01415856>**

Submitted on 15 Dec 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# DECENTRALIZED PROPORTIONAL LOAD BALANCING

J. ANSELMi\* AND N.S. WALTON†

**Abstract.** Load balancing is a powerful technique commonly used in communication and computer networks to improve system performance, robustness and fairness. In this paper, we consider a general model capturing the performance of communication and computer networks, and on top of it we propose a decentralized algorithm for balancing load among multiple network paths. The proposed algorithm is inspired by the modus operandi of the processor-sharing queue and on each network entry point operates as follows: every time a unit of load completes its service on a path, it increases by one unit the load of that path and decreases by one unit the load of a path selected at random with probability proportional to the amount of load on each of the available paths. We develop a dynamical system to argue that our load-balancer achieves a desirable network-wide utility optimization.

**Key words.** Multipath routing, decentralized load balancer, utility optimization, proportional fairness.

**AMS subject classifications.** 68M10, 68W15, 90B18

**1. Introduction.** We focus on communication and computer networks whose performance (or behavior) at equilibrium can be modelled by the solution of an utility maximization problem of the form

$$(1.1) \quad \max \sum_{r \in \mathcal{R}} n_r U_r \left( \frac{\Lambda_r}{n_r} \right) \quad \text{subject to} \quad \sum_{r: j \in r} \Lambda_r \leq C_j, \quad j \in \mathcal{J} \quad \text{over} \quad \Lambda_r \geq 0, \quad r \in \mathcal{R},$$

where  $\mathcal{R}$  denotes the set of network paths (or routes),  $\mathcal{J}$  denotes the set of network congestible resources (or links, queues),  $C_j$  denotes the service speed (or capacity) of link  $j$ ,  $n_r$  denotes the number of packets (or customers) using route  $r$ ,  $U_r$  is a strictly-concave function, and the decision variable  $\Lambda_r$ ,  $r \in \mathcal{R}$ , is interpreted as the *throughput* given by the network to all packets moving on route  $r$ , i.e. the rate at which packets flow on route  $r$ . The value of function  $U_r(\cdot)$  is interpreted as the *utility* of a packet sent on route  $r$ . The constraints of such optimization can be interpreted as the capacity constraints or the *stability region* of the network and the form of the utility functions depends on which dynamics or communication protocols packets are forced to follow. Furthermore, within the time scale of interest, the numbers of packets circulating on each route  $n_r$ ,  $r \in \mathcal{R}$ , is considered constant.

Knowing the throughput of packets on each route  $r$ , that is the unique optimizer of (1.1), one immediately obtains other performance metrics of interest such as the delay experienced by each packet on  $r$ , which follows by an application of Little's law [24]. In the following, we say that the performance (or behavior) of a network is known if the per-route throughputs of that network are known. We will also refer to (1.1) as the *single-path utility optimization*.

The utility maximization (1.1) is known to provide a general model able to approximate the performance achieved by a wide class of both communication and computer

---

\*Basque Center for Applied Mathematics (BCAM), Al. de Mazarredo 14, 48009 Bilbao, Basque Country, Spain; INRIA Bordeaux Sud Ouest, 200 av. de la Vieille Tour, 33405 Talence, France (jonatha.anselmi@inria.fr)

†University of Amsterdam, Science Park 904, 1098 XH, Amsterdam, The Netherlands (n.s.walton@uva.nl)

networks. In the networking community, it is well accepted that the throughputs achieved by packets in telecommunication networks that implement a form of congestion control, e.g. TCP (Transmission Control Protocol), can be approximated by the unique optimizer of an optimization of the form (1.1) [13, 27, 43]. Similarly, in computer science, the throughputs achieved by customers in data-centers are often modelled by the throughputs of a multiclass BCMP or Kelly queueing network [36, 1], and it is known that the throughputs of such queueing network can be approximated again by the unique optimizer of an optimization of the form (1.1) [33, 39, 2].

Keeping (1.1) as a general framework to model the behavior of communication and computer networks, we are interested in the problem of injecting a form of decentralized load-balancing control for the packets. This can be done by allowing each network entry point to send packets among all of its available routes. Such approach is motivated by the natural fact that communication and computer network resources can send and receive data from multiple paths simultaneously, and that developing such control is currently a desirable feature of strong interest, as it aims at improving performance, fairness and robustness of the network by better exploiting its intrinsic path diversity. There is a vast literature on this subject and we point the reader to Section 1.1 for an overview of the related work. In several cases of practical interest, it is desirable to design a decentralized load-balancing control with the property that the resulting throughputs converge, at equilibrium, to a solution of an utility maximization problem of the form

$$\begin{aligned}
(1.2a) \quad & \text{maximize} && \sum_{s \in \mathcal{S}} w_s U_s(\Gamma_s / w_s) \\
(1.2b) \quad & \text{subject to} && \Gamma_s = \sum_{r \in s} \Lambda_r, && s \in \mathcal{S} \\
(1.2c) \quad & && \sum_{r: j \in r} \Lambda_r \leq C_j, && j \in \mathcal{J} \\
(1.2d) \quad & \text{over} && \Gamma_s, \quad \Lambda_r \geq 0, && s \in \mathcal{S}, r \in \mathcal{R},
\end{aligned}$$

where  $\mathcal{S}$  denotes the set of network source-destination pairs,  $w_s = \sum_{r \in s} n_r$  denotes the total number of packets using all routes belonging to  $s$ , and the remainder of the notation is as above. Thus, the load-balancer is distributed among all the source-destination pairs  $s$ , and on each  $s$  it aims at maximizing the utility of the total throughput among all the available routes. This is often referred to as resource pooling. Here the individual pathwise constraints of (1.1) are combined and throughput is optimized over different source-destination pairs. There are several desirable properties in designing a decentralized load-balancer achieving the optimization (1.2); see [16, 31]. For instance, the optimizers of (1.2) can be interpreted as a Nash or Wardrop equilibrium where the delays on the routes of a given source-destination are all equal and minimum. Due to its utilitarian objective, throughputs achieved by (1.2) are fair. We will refer to (1.2) as the *multipath utility optimization*.

**1.1. Related work.** During the last decades, decentralized load balancing has been a central theme for different research areas. The objective is to improve the performance, robustness and fairness of a network by better exploiting its intrinsic path diversity. In data-centers, for instance, web or application servers are usually replicated several times [36], are often multi-homed [30] and, due to their large sizes, the network structure has several entry points. In telecommunication networks, smartphones can simultaneously use 3G and WiFi [28], and end hosts of peer-to-peer net-

works can usually choose among different download sources, e.g. as in BitTorrent, [8]. This recurrent network structure is ideal for developing decentralized load-balancing schemes.

In the networking community, researchers have recently implemented a multipath version of the TCP (Transmission Control Protocol) [41, 6], and a myriad of (decentralized) load-balancing congestion-control algorithms have been developed during the last decade; see, e.g., [14, 9, 42, 10, 40, 16, 5, 29, 17, 22]. Here an objective is to allocate bandwidth that solves a version of the multipath utility optimization (1.2). Necessary and sufficient conditions for achieving the multipath utility optimization are developed in [29]. Convergence to the solution of the multipath utility optimization is usually proven through models based on ordinary differential equations.

**1.2. Our Contribution.** On top of the general model for network performance (1.1), we propose a decentralized algorithm that balances the amount of packets associated to each network route. On each entry point of the network, our load balancer only uses local information and works as follows:

*Every time a packet completes on a route:*

- i) increase the number of packets on that route by one;*
- ii) remove one packet from a route that is selected with probability proportional to the number of packets on that route.*

Underlying this algorithm, there is the heuristic argument that the dynamics of the number of packets on each route should mimic the dynamics of a Processor-Sharing (PS) queue<sup>1</sup> [18] (from the standpoint of each network resource that implements such control). This analogy with the PS queue is intuitively driven by the following coupling argument. When our load balancer increases by one the number of packets in the route where the completion occurred, this corresponds to an arrival event at the processor-sharing queue. Consequently, a packet removal from its route is interpreted as a service event at the queue and, as in the processor-sharing queue, this event occurs at a rate that is proportional to the number of packets in each route.

Using differential equations, we develop a mathematical model describing the evolution of the number of packets on each route that is induced by our load balancer. As we describe in Section 3.2, our model assumes that between any two consecutive load-balancing updates, i.e. steps *i*) and *ii*) above, performed on any source-destination  $s$ , the network achieved the (equilibrium) throughputs given by the single-path utility optimization (1.1). Though we argue that our qualitative results hold even without it, this time-scale separation argument helps us in reducing the complexity of the underlying mathematical analysis and more importantly allows for an interpretation of the resulting equations. In fact, we will exhibit a Lyapunov function  $L(t)$  for the proposed dynamical system satisfying the following property:

$$\frac{dL}{dt} = - \sum_{s \in S} \Gamma_s D\left(\left(\frac{\Lambda_r}{\Gamma_s}\right)_{r \in s} \parallel \left(\frac{n_r}{w_s}\right)_{r \in s}\right) - \sum_{s \in S} \Gamma_s D\left(\left(\frac{n_r}{w_s}\right)_{r \in s} \parallel \left(\frac{\Lambda_r}{\Gamma_s}\right)_{r \in s}\right),$$

where  $D(\cdot \parallel \cdot)$  denotes the Kulback-Liebler divergence or relative entropy. From this, we understand the effect of our processor-sharing heuristic: our load-balancer can be interpreted as to be equilibrating the relative entropy between the distribution of packets along each route and the distribution of throughputs along these routes. This entropy approach contrasts previous studies of multipath routing, which tend to focus

<sup>1</sup>In a PS queue, all jobs are processed simultaneously and each of them takes  $1/n$  of the available capacity, provided that  $n$  denotes the number of jobs in the queue.

on placing quadratic bound on Lyapunov functions [14, 16] and/or applying Nyquist's stability criterion [9].

Under proportional fairness [13], i.e.  $U_r(\Lambda_r/n_r) = \log(\Lambda_r/n_r)$ , we prove in our main result that the throughputs achieved by the packets converge to the solutions of the multipath utility optimization (1.2). Our proof is essentially based on duality arguments connecting the structure of the Lyapunov function  $L(t)$  to the objective of the multipath utility optimization. Then, we adapt our approach to show that it is possible to design an other load-balancer able to achieve the multipath utility optimization under a general class of utility functions. This follows through an optimization decomposition argument that is similar to the one used in [12].

The remainder of the paper is organized as follows. Section 2 introduces the modeling framework. Section 3 presents our decentralized load-balancer, describes application areas and develops a mathematical model for the dynamics of network packets. Section 4 is devoted to our main results (Theorems 4.2 and 4.4), which are proven in Section 5. A number of technical lemmas are deferred to the Appendix for simplicity of exposition.

**2. Multipath Utility Optimization.** Let the set  $\mathcal{J}$ , indexed by  $j$ , be the set of *links*. These are the network resources affected by congestion and, depending on the context, can be also interpreted as queues, Internet routers, or web-servers. Packets circulate in the network following given *routes*. A route is an ordered set of links,  $r = (j_1, \dots, j_{k_r})$  with  $j_1, \dots, j_{k_r} \in \mathcal{J}$ . Thus,  $k_r$  is the number of links on route  $r$ . The set  $\mathcal{R}$  is the set of routes. A *source-destination*,  $s$ , is a set of routes, thus  $s \subset \mathcal{R}$ . Each source-destination contains the set of routes available to a *host*. We think of hosts as the entry points of the network and can be also interpreted as dispatchers or end users. The set  $\mathcal{S}$  gives the set of source-destinations. We assume that there are no source-destinations that share a common route, i.e.,  $s \cap s' = \emptyset$  when  $s \neq s'$ . Thus, we can define  $s(r) \in \mathcal{S}$  as the source-destination to which route  $r$  belongs. We assume that there is one host for each source-destination.

Let  $n_r$  denote the number of packets on route  $r \in \mathcal{R}$ . Correspondingly,  $n = (n_r : r \in \mathcal{R})$  is the number of packets on each route. Let  $w_s$  denote the number of packets on source-destination  $s \in \mathcal{S}$ . Accordingly,  $w = (w_s : s \in \mathcal{S})$  is the number of packets on each source-destination. Thus, for  $s \in \mathcal{S}$

$$(2.1) \quad w_s = \sum_{r \in s} n_r.$$

The strictly positive vector  $C = (C_j : j \in \mathcal{J})$  gives the *capacity* (or *service rate*) of each link, i.e. the number of packets served per unit of time in a fully utilized link. Let  $\Lambda_r$  denote the *throughput* (or *flow rate* or *bandwidth*) allocated by the network to the packets on route  $r \in \mathcal{R}$ , and let  $\Gamma_s$  denote the throughput on source-destination  $s \in \mathcal{S}$ . Thus, for  $s \in \mathcal{S}$  and  $j \in \mathcal{J}$

$$\Gamma_s = \sum_{r \in s} \Lambda_r, \quad \sum_{r: j \in r} \Lambda_r \leq C_j.$$

**2.1. Proportional Fairness.** When a host *can* distribute packets among multiple routes, one wishes to design a load-balancer maximizing the utility of the total throughput between each source-destination pair, and thus achieving the multi-path utility maximization (1.2). We will first assume that both the optimizations (1.1) and (1.2) have logarithmic utilities, i.e.

$$(2.2) \quad U_r(\Lambda_r/n_r) = \log(\Lambda_r/n_r), \quad U_s(\Gamma_s/w_s) = \log(\Gamma_s/w_s),$$

for  $r \in \mathcal{R}$  and  $s \in \mathcal{S}$  respectively. The throughputs achieved with this utility function are called *proportionally fair*. Several economic and game-theoretic interpretations hold in this case [12, 13, 25, 35, 23, 43]. Communication protocols achieving this optimization objective have been designed and implemented [26, 37, 20]. Furthermore, proportionally fair throughputs approximate the performance of closed multi-class BCMP or Kelly queueing networks [34, 39, 2], which have been used to model the performance of data centers [36].

Summarizing so far, our primary goal is to design a decentralized load-balancer ensuring that the resulting throughputs solve

$$(2.3a) \quad \max_{\Gamma, \Lambda \geq 0} \sum_{s \in \mathcal{S}} w_s \log \Gamma_s \quad \text{s.t.} \quad \Gamma_s = \sum_{r \in s} \Lambda_r, \quad s \in \mathcal{S}, \quad \sum_{r: j \in r} \Lambda_r \leq C_j, \quad j \in \mathcal{J}.$$

We refer to problem (2.3) as the *multipath proportionally fair optimization*.

We will achieve this optimization by considering a simple load-balancing mechanism which modifies the *single-path proportionally fair optimization*:

$$(2.4a) \quad \max_{\Lambda \geq 0} \sum_{r \in \mathcal{R}} n_r \log \Lambda_r \quad \text{s.t.} \quad \sum_{r: j \in r} \Lambda_r \leq C_j, \quad j \in \mathcal{J}.$$

**3. Processor Sharing Load Balancing.** We develop a simple decentralized algorithm that balances the amount of packets among multiple routes of a network. Operated at each host, it sends incoming packets along a given route with a rate proportional to the number of packets in transfer on that route. This modus operandi is similar to how a Processor-Sharing (PS) queue serves different packet classes [18]. The proportion of service devoted to a class in a PS queue is proportional to the number of jobs in that class. Given this analogy, we refer to our algorithm as “Processor Sharing Load Balancer” (PSLB), which is as follows:

*Every time a packet completes on a route:*

- i) increase the number of packets on that route by one;*
- ii) then, remove one packet from a route that is selected with probability proportional to the number of packets on that route.*

It is clear that PSLB is easy to implement on each host, which only needs to store the amount of packet on each of its routes. Note that the probability of decreasing the number of packets on each route is proportional to the number of packets on each route, as in a PS queue.

**3.1. Application areas.** There exist at least four scenarios where our algorithm can be implemented: multipath-TCP, data-centers, wireless networks and peer-to-peer networks. We discuss each of these.

*i) Multipath-TCP:* Multipath-TCP (MPTCP) refers to a set of extensions of TCP (Transmission Control Protocol) that allow each connection to use multiple paths simultaneously. MPTCP allows smartphones to simultaneously use 3G and WiFi [28]; it reacts to link failures on a time-scale that is much quicker than that obtained by updating the routing tables of network routers; it increases resource pooling – that is where multiple network resources can collectively be seen as a single *pooled* resource [21, 7]. Due to these potential benefits, the Internet Engineering Task Force (IETF) MPTCP working group has recently released three implementations of MPTCP [32]. The first large-scale commercial deployment of MPTCP appeared on September 18th,

2013, when Apple released iOS 7 [4]. In this setting, our approach aims at balancing the magnitude of the congestion window associated to each network route available to host  $s$  given the total congestion window  $w_s$ .

*ii) Data Centers:* In a data-center, packets are interpreted as http requests, links as Internet servers and each source-destination as the set of routes available to execute a web-application. In practice, end-users submit an http request, read on a browser the result that has been produced by the data-center, and then submit again an other request. Http requests can be assumed to come from a constant number of end-users, and the network performance can be modelled by a closed multiclass BCMP/Kelly queueing network; see, e.g., [36, 1] and references therein. Given their large size, data-centers usually admit multiple entry points, called dispatchers, which can implement a load-balancing mechanism. By doing so and assuming that throughputs without load-balancing can be modelled by an optimization of the form (1.1) (as in, e.g., [1]), we show that our load-balancer will make throughputs converge to a solution of an optimization of the form (1.2).

*iii) Wireless Networks.* In wireless systems, there are reasons to balance load amongst multiple resources. First, hosts may have access to multiple frequencies in the wireless spectrum and these are usually shared between users. Thus, load balancing is highly relevant in this dynamic spectrum access setting. Further, a wireless host may have access to multiple transmitters whose signal quality will, amongst other factors, depend on the distance between host and the transmitter. This access point selection problem represents a further model where load balancing is required. For an analogous proportional fair load-balancing system and discussion on these systems, see [44].

*iv) Peer-to-peer networks.* Although we refer to transfers between source destination pairs, it is clear from the above examples that a single-source is not required in our modeling setting. The case of peer-to-peer networks gives a further example. Here, a peer may download elements of a file from multiple peers. By exploiting the diversity of peers who share a specific file, users can receive high and fair download rates. Once again, the traffic must be balanced between multiple routes and thus load-balancing techniques are of importance, see [11].

**3.2. Differential Equation Model.** We model the time evolution of the number of packets induced by PSLB through differential equations. Between two consecutive load-balancing updates, we assume that the throughputs allocated to packets by the network are close to  $\Lambda(n)$ , i.e., the equilibrium rates achieved by the single-path optimization (1.1). We stress that this is possible because rates converge to such single-path optimization. In other words, we assume that our load-balancer operates at a time-scale that allows the single-path equilibrium rates to be achieved.

A packet completion on a route  $r \in s$  will result in a increase in the number of packets on that route. In view of the time-scale separation argument above, the rate allocated by the network to such completions is  $\Lambda_r(n)$ . Each completion on a source-destination results in a decrease in the congestion of a route with probability proportional to the number of packets on that route. These source-destination completions occur at rate  $\Gamma_s(n)$  and the probability of decrease on route  $r$  is  $\frac{n_r}{w_s}$ . Thus, we model the time evolution of the number of packets in each route obtained when PSLB is used by the following system of ordinary differential equations: for  $n_r(t) \geq 0$

$$(3.1) \quad \frac{dn_r}{dt} = \Lambda_r(n) - \Gamma_s(n) \frac{n_r}{w_s},$$

for all  $r \in s$  and for all  $s \in \mathcal{S}$ , where  $n(0) \geq 0$  is such that  $\sum_{r \in s} n_r(0) = w_s$  for all  $s$ . We refer to any solution  $n(t)$  of (3.1) as a *fluid solution*.

When each source-destination contains a single route and the total number of packets is kept constant, our model can be compared to the fluid-limit of the flow-level model studied in [3], which was proposed to model the time evolution of connections in a communication network with singlepath congestion control.

Before continuing, we observe that a special case of practical interest is a network of parallel servers, which gives a bipartite network structure between hosts and servers. The applications of wireless networks and data centers discussed above are two examples of a parallel server system. In this case, the solution to the proportional fair optimization takes a particularly simple form. Namely, the throughput at route  $r$  is proportional to the number of packets (or jobs) that have been sent there, i.e.,

$$\Lambda_r(n) = \frac{n_r}{\sum_{r': j \in r'} n_{r'}} C_j,$$

where  $j \in r$ . In this case, the above differential equation takes a very simple form.

**4. Main Convergence Results.** Using the differential equation model developed in previous section, we prove our main results (Theorems 4.2 and 4.4). In Theorem 4.2, we prove that the throughputs achieved by PSLB converge to a solution of the multipath proportional fair optimization (2.3). In Theorem 4.4, we show that these properties can be also achieved under a wide class of utility functions.

**4.1. Multipath Proportional Fair Optimization.** We will demonstrate that our load balancer is able to modify the single-path flows, which solve the proportionally fair optimization, into a multipath flow control, which solves the multipath proportional fair optimization. For this purpose, we use the following function

$$(4.1) \quad L(n) \stackrel{\text{def}}{=} \sum_{s \in \mathcal{S}} \sum_{r \in s} n_r \log \left( \frac{n_r}{w_s \Lambda_r(n)} \right).$$

Letting  $L^* \stackrel{\text{def}}{=} \min_{n \geq 0} L(n)$  subject to  $\sum_{r \in s} n_r = w_s$  for all  $s$ , we show that  $L(n) - L^*$  is a Lyapunov function for the dynamical system (3.1).

The crucial observation about this function is given in the following proposition.

**THEOREM 4.1.** *For each point of differentiability of a fluid solution  $n(t)$ , the following holds*

$$(4.2a) \quad \frac{dL(n(t))}{dt} = - \sum_{s \in \mathcal{S}} \Gamma_s(n) \left[ D \left( \frac{(\Lambda_r(n))_{r \in s}}{\Gamma_s(n)} \middle\| \frac{(n_r)_{r \in s}}{w_s} \right) + D \left( \frac{(n_r)_{r \in s}}{w_s} \middle\| \frac{(\Lambda_r(n))_{r \in s}}{\Gamma_s(n)} \right) \right],$$

where, for two probability distributions  $p^s = (p_r : r \in s)$  and  $q^s = (q_r : r \in s)$ ,  $D(p^s || q^s)$  defines the relative entropy

$$(4.3) \quad D(p^s || q^s) = \sum_{r \in s} p_r \log \frac{p_r}{q_r}.$$

Note that for two discrete probability distributions  $p$  and  $q$  on the same support, it is well known, using Jensen's inequality, that  $D(p || q) \geq 0$  and that  $D(p || q) = 0$  if and only if  $p = q$ . This immediately implies that  $L(n(t))$  is non-increasing.

**REMARK 1.** *The symmetric relative entropy*

$$D(p^s || q^s) + D(q^s || p^s),$$



as first introduced by Kullback and Leibler [19], is critical to understanding the convergence of single-path connections to the multi-path optimum. Through the symmetric relative entropy (4.2), PSLB attempts to equate the distribution of packets within the network and the distribution of throughputs along different routes. If these are unequal then the Lyapunov function will decrease further towards its minimum. Through duality arguments presented shortly, we see that the minimum of the Lyapunov function  $L(n)$  is achieved only at solutions which minimize the multipath proportional fair optimization.

As a consequence of the last theorem, we obtain our main result, which says that dynamics of (3.1) converge to the multipath proportional fair optimization (2.3).

**THEOREM 4.2.** *For any fluid solution for which  $L(n(t))$  is absolutely continuous, the source-destination throughputs converge to  $\Gamma^*$  the optimal solution to the multipath proportionally fair optimization (2.3),*

$$\Gamma(n(t)) \xrightarrow[t \rightarrow \infty]{} \Gamma^*.$$

A proof of Theorem 4.1 and Theorem 4.2 can be found in Section 5.1.

**4.2. Multipath Utility Optimization.** Our prior analysis achieved the multipath proportional fair objective. However, in this section we show that it can be generalized to a wide class of utility functions. Through an optimization decomposition argument, it is known that other utility optimizations can be achieved by altering the weights associated with the proportional fair objective [12]. Now, we apply a similar decomposition in order to modify proportional fair single-path flows towards a general multipath utility optimization.

First, we introduce some additional notation. For each source destination pair, let  $U_s : \mathbb{R}_+ \rightarrow \mathbb{R}$  be a strictly-increasing continuously-differentiable utility function. We further assume that the function  $\Gamma_s \mapsto \Gamma_s U'_s(\Gamma_s)$  is continuous, strictly increasing and with range  $(0, \infty)$ . We note that this assumption holds for all the weighted  $\alpha$ -fair allocations with  $\alpha < 1$  [27]. Given this assumption, we define the function  $g_s(w_s) = \Gamma_s$  such that

$$(4.4) \quad g_s^{-1}(\Gamma_s) = \Gamma_s U'_s(\Gamma_s),$$

which is well-defined since  $\Gamma_s U'_s(\Gamma_s)$  is increasing.

We interpret  $g_s(w_s)$  as the throughput source-destination pair  $s$  would expect given that it has utility function  $U_s$  and given that its current number of packets from source to destination is  $w_s$ . From this interpretation, we now consider a load balancer who obeys the following dynamics

$$(4.5) \quad \frac{dn_r}{dt} = \Lambda_r(n) - g_s(w_s) \frac{n_r}{w_s},$$

where  $\Lambda_r(n)$  is the solution to the single-path proportionally fair optimization problem (2.4). In view of the fluid model (3.1), we note that these rates are a form of processor-sharing load balancing: once again, we increment the number of packets for each packet arrival and decrement it proportionally to the number of packets present in each of the available routes. The only difference to our prior analysis is that the number of packets decreases at rate  $g_s(w_s)$  rather than  $\Gamma_s$ , so the number of packets between each source-destination pair is no longer conserved. Summing over  $r \in s$ , we

see that  $w_s$  must vary dynamically according to the equations

$$\frac{dw_s}{dt} = \Gamma_s(n) - g_s(w_s), \quad \text{where} \quad \Gamma_s(n) = \sum_{r \in s} \Lambda_r(n), \quad w_s = \sum_{r \in s} n_r.$$

We want to control the number of packets between each source destination pair so that the logarithm in the proportionally fair optimization is replaced by the more general utility function  $U_s$ . With this in mind, we consider the function

$$(4.6) \quad G_s(w_s) = \max_{\Gamma_s \geq 0} \{w_s \log \Gamma_s - U_s(\Gamma_s)\}.$$

With the addition of function  $G_s$  and motivated by the structure of (4.1), we define the following function

$$(4.7) \quad H(n) = \sum_{r \in \mathcal{R}} n_r \log \frac{n_r}{w_s \Lambda_r(n)} + \sum_{s \in \mathcal{S}} G_s(w_s).$$

Letting  $H^* \stackrel{\text{def}}{=} \min_{n \geq 0} H(n)$  subject to  $\sum_{r \in s} n_r = w_s$  for all  $s$ , we show that  $H(n) - H^*$  is a Lyapunov function for the dynamical system (4.5).

Once again, the function  $H$  balances the relative entropy between different routes of the network. This is shown in the following proposition, which is proven in Section 5.2.

**THEOREM 4.3.** *For each point of differentiability of a fluid solution for (4.5),  $n(t)$ , the following holds*

$$(4.8) \quad \begin{aligned} \frac{dH(n(t))}{dt} = & - \sum_{s \in \mathcal{S}} \Gamma_s D\left(\frac{(\Lambda_r)_{r \in s}}{\Gamma_s} \middle| \middle| \frac{(n_r)_{r \in s}}{w_s}\right) - \sum_{s \in \mathcal{S}} g_s(w_s) D\left(\frac{(n_r)_{r \in s}}{w_s} \middle| \middle| \frac{(\Lambda_r)_{r \in s}}{\Gamma_s}\right) \\ & - \sum_{s \in \mathcal{S}} \left(\Gamma_s - g_s(w_s)\right) \left(\log \Gamma_s - \log g_s(w_s)\right). \end{aligned}$$

Using that the logarithm is increasing, the term (4.8) is negative. Furthermore, since  $D(\cdot | \cdot) \geq 0$ , this means that the derivative of the entropy function  $H(n(t))$  is negative.

As a consequence of Theorem 4.3 and adapting the arguments in the proof of Theorem 4.2, we can prove that the single-path flows converge to solutions of the multipath utility optimization (1.2).

**THEOREM 4.4.** *For any fluid solution for which  $H(n(t))$  is absolutely continuous and  $w(t)$  is bounded from zero, the source-destination throughputs converge to  $\Gamma^*$  the optimal solution to the multipath utility optimization (1.2),*

$$\Gamma(n(t)) \xrightarrow[t \rightarrow \infty]{} \Gamma^*.$$

A proof of Theorem 4.3 and 4.4 can be found in Section 5.2.

**5. Proofs of Main Results.** We present proofs for our main results: Theorem 4.2 and Theorem 4.4. For simplicity, we often drop the dependency on  $n$  when referring to quantities  $\Lambda_r(n)$  and  $\Gamma_s(n)$ .

**5.1. Proof of Theorem 4.1 and Theorem 4.2.** We require the following lemma which follows by somewhat standard calculus arguments, for instance see [38, Lemma 2] for a proof.

LEMMA 5.1. *If  $n_r > 0$  then*

$$\frac{\partial L(n)}{\partial n_r} = \log \left( \frac{n_r}{w_s \Lambda_r(n)} \right) + 1.$$

With this, we can prove Theorem 4.1.

*Proof.* [Proof of Theorem 4.1] For the following sequence of equalities, we use the shorthand that  $n = n(t)$ ,  $L = L(n(t))$  and  $\Gamma = \Gamma(n(t))$ :

$$\begin{aligned} \frac{dL}{dt} &= \sum_{r \in \mathcal{R}} \frac{dn_r}{dt} \frac{\partial L}{\partial n_r} = \sum_{s \in \mathcal{S}} \sum_{\substack{r \in \mathcal{S} \\ n_r > 0}} \left( \Lambda_r - \frac{n_r}{w_s} \Gamma_s \right) \left( \log \frac{n_r}{w_s \Lambda_r} + 1 \right) \\ (5.1a) \quad &= \sum_{s \in \mathcal{S}} \sum_{r \in \mathcal{S}} \left( \Lambda_r - \frac{n_r}{w_s} \Gamma_s \right) \log \frac{n_r}{w_s \Lambda_r} \\ (5.1b) \quad &= - \sum_{s \in \mathcal{S}} \Gamma_s \sum_{r \in \mathcal{S}} \frac{\Lambda_r}{\Gamma_s} \log \frac{w_s \Lambda_r}{n_r \Gamma_s} - \sum_{s \in \mathcal{S}} \Gamma_s \sum_{r \in \mathcal{S}} \frac{n_r}{w_s} \log \frac{n_r \Gamma_s}{w_s \Lambda_r} \\ (5.1c) \quad &= - \sum_{s \in \mathcal{S}} \Gamma_s D \left( \frac{(\Lambda_r)_{r \in \mathcal{S}}}{\Gamma_s} \middle| \middle| \frac{(n_r)_{r \in \mathcal{S}}}{w_s} \right) - \sum_{s \in \mathcal{S}} \Gamma_s D \left( \frac{(n_r)_{r \in \mathcal{S}}}{w_s} \middle| \middle| \frac{(\Lambda_r)_{r \in \mathcal{S}}}{\Gamma_s} \right) \end{aligned}$$

In the first two equalities above, we apply the chain rule and the definition of the derivatives of  $n$  and  $L$ , from fluid equations (3.1) and Lemma 5.1. In the equality (5.1a), we cancel terms and observe that we can exclude terms where  $n_r = 0$  because  $\frac{dn_r}{dt} = 0$  and  $\Lambda_r(n) = 0$  must hold at times when  $n_r = 0$ . Then, in equalities (5.1b) and (5.1c) we add and subtract terms  $\Gamma_s \log \Gamma_s$  and observe that the terms summed are relative entropy terms.  $\square$

To prove Theorem 4.2, we will need the following technical lemma, which is straightforward to verify and gives first order properties for our Lyapunov function  $L$ .

LEMMA 5.2.  *$L(n)$  is continuous and convex and thus  $\mathcal{N}^*$ , the set of the optimal solutions to (5.1), is compact and convex.*

A proof of Lemma 5.2 can be found in the Appendix. The following Lemma bounds terms  $\Gamma_s(n)$ .

LEMMA 5.3. *a) The function  $L(n)$  can be characterized through its dual as follows*

$$(5.2) \quad L(n) = \max_{q \geq 0} \sum_{r \in \mathcal{R}} n_r \log \left( \sum_{j \in \mathcal{R}} q_j \right) - \sum_{j \in \mathcal{J}} q_j C_j + \sum_{r \in \mathcal{R}} n_r - \sum_{s \in \mathcal{S}} w_s \log w_s.$$

*b) There exists a constant  $\Gamma_{\min} > 0$  not dependent on  $n$  such that for all  $n$  and each  $s \in \mathcal{S}$  we have  $\Gamma_s(n) \geq \Gamma_{\min}$ .*

A proof of Lemma 5.3 can be found in the Appendix.

The following lemma finds a dual description for the Lyapunov function  $L$  and also equates the minimized Lyapunov function with the multipath proportionally fair optimization.

LEMMA 5.4. *The minimum of Lyapunov function  $L(n)$  relates to the multi-path proportionally fair optimization as follows*

$$\min_{n \geq 0} L(n) \quad \text{subject to} \quad \sum_{r \in \mathcal{S}} n_r = w_s, \quad s \in \mathcal{S}$$

$$= - \max_{\Gamma, \Lambda \geq 0} \sum_{s \in \mathcal{S}} w_s \log \Gamma_s \quad \text{subject to} \quad \Gamma_s = \sum_{r \in s} \Lambda_r, \quad s \in \mathcal{S}.$$

Moreover,  $L(n)$  is minimal if and only if

$$(5.3) \quad \frac{n_r}{w_s} = \frac{\Lambda_r(n)}{\Gamma_s(n)}$$

for each  $r \in \mathcal{R}$  (with  $r \in s$ ) such that  $n_r > 0$ .

*Proof.* We have the following equalities

$$\begin{aligned} & \min_{n \geq 0} L(n) \quad \text{subject to} \quad \sum_{r \in s} n_r = w_s, \quad s \in \mathcal{S} \\ &= \min_{n \geq 0} \min_{\Lambda \geq 0} \sum_{s \in \mathcal{S}} \sum_{r \in s} n_r \log \frac{n_r}{\Lambda_r w_s} \quad \text{s.t.} \quad \sum_{r: j \in r} \Lambda_r \leq C_j, \quad \sum_{r \in s} n_r = w_s, \quad j \in \mathcal{J}, s \in \mathcal{S} \\ &= \min_{\Lambda \geq 0} \min_{n \geq 0} \sum_{s \in \mathcal{S}} \sum_{r \in s} n_r \log \frac{n_r}{\Lambda_r w_s} \quad \text{s.t.} \quad \sum_{r \in s} n_r = w_s, \quad \sum_{r: j \in r} \Lambda_r \leq C_j, \quad j \in \mathcal{J}, s \in \mathcal{S} \\ &= \min_{\Lambda \geq 0} \min_{n \geq 0} \sum_{s \in \mathcal{S}} w_s \sum_{r \in s} \frac{n_r}{w_s} \log \left( \frac{n_r}{w_s} \frac{\Gamma_s}{\Lambda_r} \right) - \sum_{s \in \mathcal{S}} w_s \log \Gamma_s \\ & \quad \text{subject to} \quad \sum_{r \in s} n_r = w_s, \quad \sum_{r: j \in r} \Lambda_r \leq C_j, \quad \sum_{r \in s} \Lambda_r = \Gamma_s \\ &= \min_{\Lambda \geq 0} - \sum_{s \in \mathcal{S}} w_s \log \Gamma_s \quad \text{s.t.} \quad \sum_{r: j \in r} \Lambda_r \leq C_j, \quad \sum_{r \in s} \Lambda_r = \Gamma_s \end{aligned}$$

The first equality follows by the definition of  $L(n)$ ; in the second equality, we interchange the order of minimization; in the third, we add and subtract the multipath proportionally-fair objective; in the final equality, we minimize over  $n$  and observe that the relative entropy term in the objective is minimized to zero when the vector  $n$  is proportional to  $\Lambda$ .

Now we show that (5.3) is a condition for optimality of  $L(n)$ . Consider the function

$$(5.4) \quad \tilde{L}(n, \Lambda) = \sum_{s \in \mathcal{S}} \sum_{r \in s} n_r \log \frac{n_r}{\Lambda_r w_s}.$$

It is clear that  $\tilde{L}(n, \Lambda)$  is convex. The minimum of  $\tilde{L}(n, \Lambda)$  over  $\Lambda$  (satisfying the network capacity constraints  $\sum_{r: j \in r} \Lambda_r \leq C_j, j \in \mathcal{J}$ ) is achieved at  $\Lambda(n)$  and the minimum attained at  $L(n)$ . The minimum of  $\tilde{L}(n, \Lambda)$  over  $n$  (satisfying the packet constraints  $\sum_{r \in s} n_r = w_s$ ), is achieved when  $n_r \propto \Lambda_r$ . Since we are minimizing a convex function over a product space, its minimum is the intersection of the marginal minimality conditions, i.e.  $\tilde{L}(n, \Lambda)$  and thus  $L(n)$  is minimized iff  $\Lambda_r = \Lambda_r(n)$ , for  $r \in \mathcal{R}$  and, for all  $r \in s$ ,

$$\frac{\Lambda_r(n)}{\Gamma_s(n)} = \frac{n_r}{w_s}.$$

□

The proof of Theorem 4.2 is now fairly straight-forward.

*Proof.* [Proof of Theorem 4.2] We let  $L^*$  be the minimal value of  $L(n)$  over  $n$  with  $\sum_{r \in \mathcal{S}} n_r = w_s$ ,  $s \in \mathcal{S}$ . The last lemma states that

$$(5.5) \quad L(n) = L^* \quad \text{iff} \quad \frac{n_r}{w_s} = \frac{\Lambda_r(n)}{\Gamma_s(n)}.$$

From Proposition 4.1, we have that

$$\frac{dL}{dt} = - \sum_{s \in \mathcal{S}} \Gamma_s D \left( \frac{(\Lambda_r)_{r \in s}}{\Gamma_s} \left\| \frac{(n_r)_{r \in s}}{w_s} \right\| \right) - \sum_{s \in \mathcal{S}} \Gamma_s D \left( \frac{(n_r)_{r \in s}}{w_s} \left\| \frac{(\Lambda_r)_{r \in s}}{\Gamma_s} \right\| \right).$$

Thus, as the relative entropy  $D(p||q) = 0$  iff  $p = q$ , we can combine the last two displays to give that

$$(5.6) \quad \frac{dL}{dt} = 0 \quad \text{iff} \quad \frac{\Lambda_r(n)}{\Gamma_s(n)} = \frac{n_r}{w_s}, \quad r \in s, s \in \mathcal{S} \quad \text{iff} \quad L(n) = L^*.$$

So  $L(n(t))$  is decreasing unless  $n(t)$  achieves the minimum of  $L(n)$ . Suppose that  $L(n(t))$  decreases to some value  $L'$  (which may or may not equal  $L^*$ ). There must be a sequence of times  $\{t_k\}_{k \in \mathbb{N}}$  such that  $\frac{dL(t_k)}{dt} \rightarrow 0$  as  $t_k \rightarrow \infty$ . By Lemma 5.3b, it can not be that  $\Gamma_s(n(t_k)) \rightarrow 0$  as  $k \rightarrow \infty$ , so for this sequence of times it must be that

$$(5.7) \quad D \left( \frac{(n_r(t_k))_{r \in s}}{w_s} \left\| \frac{(\Lambda_r(t_k))_{r \in s}}{\Gamma_s(t_k)} \right\| \right) \xrightarrow{k \rightarrow \infty} 0.$$

Thus, by continuity of the relative entropy at 0, any limit point,  $n^*, \Lambda^*, \Gamma^*$  of the sequences  $n(t_k), \Lambda(n(t_k)), \Gamma(n(t_k))$  must satisfy

$$\frac{\Lambda_r^*}{\Gamma_s^*} = \frac{n_r^*}{w_s},$$

for all  $r \in s, s \in \mathcal{S}$ . Thus the minimum of  $L(n(t))$  must be the optimal value  $L' = L^*$ .

So  $L(n(t))$  converges to  $L^*$ , we now show that  $\Gamma(n(t))$  converges to  $\Gamma^*$ . Observe

$$\begin{aligned} & L(n(t)) - L^* \\ &= \sum_{s \in \mathcal{S}} \sum_{r \in s} n_r(t) \log \left( \frac{n_r(t) \Gamma_s(n(t))}{w_s \Lambda_r(n(t))} \right) + \sum_{s \in \mathcal{S}} w_s \log \Gamma_s^* - \sum_{s \in \mathcal{S}} w_s \log(\Gamma_s(n(t))) \\ &\geq \sum_{s \in \mathcal{S}} w_s \log \Gamma_s^* - \sum_{s \in \mathcal{S}} w_s \log(\Gamma_s(n(t))) \\ &\geq 0, \end{aligned}$$

where in the first inequality, above, we use the positivity of the relative entropy. So since  $L(n(t)) - L^*$  converges to 0, and since  $\Gamma \mapsto \sum_s w_s \log \Gamma_s$  is continuous and  $\Gamma_s(n(t))$  is feasible for each  $t$ , it must be that, for  $s \in \mathcal{S}$ ,

$$(5.8) \quad \Gamma_s(n(t)) \xrightarrow{t \rightarrow \infty} \Gamma_s^*,$$

as required.  $\square$

**5.2. Proof of Theorem 4.3 and Theorem 4.4.** Next lemma is a consequence of Fenchel duality and relates the function  $G_s$  given in (4.6) with the utility function  $U_s$  and  $g_s$  in (4.4). The lemma also gives the derivative of our Lyapunov function.

LEMMA 5.5. *a) The function  $G_s(w_s)$  is convex. The maximization for  $G_s(w_s)$  given by (4.6) is optimized by  $g_s(w_s)$ . The function  $G_s$  is differentiable with derivative*

$$(5.9) \quad G'_s(w_s) = \log g_s(w_s).$$

Further,

$$(5.10) \quad -U_s(\Gamma_s) = \min_{w_s \geq 0} \{G_s(w_s) - w_s \log \Gamma_s\}.$$

b) The partial derivatives of  $H$  are given by the expression

$$(5.11) \quad \frac{\partial H}{\partial n_r} = \log \left( \frac{n_r}{w_s \Lambda_r(n)} \right) + \log g_s(w_s).$$

A proof of Lemma 5.5 can be found in the Appendix. With this lemma in place, we can prove Theorem 4.3.

*Proof.* [Proof of Theorem 4.3] We differentiate  $H(n(t))$  with respect to  $t$ . In the equalities below, we use the shorthand notation  $w = w(t)$ ,  $n = n(t)$ ,  $\Lambda = \Lambda(n(t))$  and  $\Gamma = \Gamma(n(t))$  for simplicity:

$$\begin{aligned} \frac{dH}{dt} &= \sum_{r \in \mathcal{R}} \frac{dn_r}{dt} \frac{\partial H}{\partial n_r} = \sum_{r \in \mathcal{R}} \left( \Lambda_r - g_s(w_s) \frac{n_r}{w_s} \right) \left( \log \left( \frac{n_r}{w_s \Lambda_r} \right) + G'_s(w_s) \right) \\ &= + \sum_{s \in \mathcal{S}} \sum_{r \in s} \Lambda_r \log \left( \frac{n_r}{w_s \Lambda_r} \right) - \sum_{s \in \mathcal{S}} \sum_{r \in s} g_s(w_s) \frac{n_r}{w_s} \log \left( \frac{n_r}{w_s \Lambda_r} \right) \\ &\quad + \sum_{s \in \mathcal{S}} \sum_{r \in \mathcal{R}} \Lambda_r G'_s(w_s) - \sum_{s \in \mathcal{S}} \sum_{r \in s} g_s(w_s) \frac{n_r}{w_s} G'_s(w_s) \\ (5.12a) \quad &= + \sum_{s \in \mathcal{S}} \Gamma_s \sum_{r \in s} \frac{\Lambda_r}{\Gamma_s} \log \left( \frac{n_r \Gamma_s}{w_s \Lambda_r} \right) - \sum_{s \in \mathcal{S}} g_s(w_s) \sum_{r \in s} \frac{n_r}{w_s} \log \left( \frac{n_r \Gamma_s}{w_s \Lambda_r} \right) \\ &\quad - \sum_{s \in \mathcal{S}} \Gamma_s \log \Gamma_s + \sum_{s \in \mathcal{S}} g_s(w_s) \log \Gamma_s + \sum_{s \in \mathcal{S}} \Gamma_s G'_s(w_s) - \sum_{s \in \mathcal{S}} g_s(w_s) G'_s(w_s) \\ (5.12b) \quad &= - \sum_{s \in \mathcal{S}} \Gamma_s D \left( \frac{(\Lambda_r)_{r \in s}}{\Gamma_s} \middle\| \frac{(n_r)_{r \in s}}{w_s} \right) - \sum_{s \in \mathcal{S}} g_s(w_s) D \left( \frac{(n_r)_{r \in s}}{w_s} \middle\| \frac{(\Lambda_r)_{r \in s}}{\Gamma_s} \right) \\ &\quad - \sum_{s \in \mathcal{S}} \left( \Gamma_s - g_s(w_s) \right) \left( \log \Gamma_s - \log g_s(w_s) \right). \end{aligned}$$

In the second equality, we used Lemma 5.5.b. In (5.12a), we have added terms  $\Gamma_s \log \Gamma_s$  and multiplied/divided by  $\Gamma_s$  to express summands in terms of the relative entropy in (5.12b). In (5.12b), we have also used (5.9).  $\square$

We let  $H^*$  be the minimal value of  $H(n)$  over  $n$  with  $\sum_{r \in s} n_r = w_s$ ,  $s \in \mathcal{S}$ . Next lemma extends Lemma 5.4.

LEMMA 5.6. *The minimum of  $H(n)$  relates to the multi-path utility optimization as follows*

$$\min_{n \geq 0, w \geq 0} H(n) \quad \text{subject to} \quad \sum_{r \in s} n_r = w_s, \quad s \in \mathcal{S}$$

$$= - \max_{\Gamma, \Lambda \geq 0} \sum_{s \in \mathcal{S}} U_s(\Gamma_s) \quad \text{subject to} \quad \Gamma_s = \sum_{r \in \mathcal{S}} \Lambda_r, \quad s \in \mathcal{S}, \quad \sum_{r: j \in r} \Lambda_r \leq C_j, \quad j \in \mathcal{J}.$$

Moreover,  $H(n)$  is minimal if and only if

$$(5.13) \quad \frac{n_r}{w_s} = \frac{\Lambda_r(n)}{\Gamma_s(n)}$$

for each  $r \in \mathcal{R}$  (with  $r \in s$ ) such that  $n_r > 0$  and

$$(5.14) \quad g_s(w_s) = \Gamma_s(n)$$

for each  $s \in \mathcal{S}$ .

A proof of Lemma 5.6 can be found in the Appendix. With Lemma 5.6 we can prove Theorem 4.4.

*Proof.* [Proof of Theorem 4.4] We let  $H^*$  be the minimal value of  $H(n)$  over  $n$ . Last lemma says that

$$(5.15) \quad H(n) = H^* \quad \text{iff} \quad g_s(w_s) = \Gamma_s(n) \quad \text{and} \quad \frac{n_r}{w_s} = \frac{\Lambda_r(n)}{\Gamma_s(n)}, \quad n_r > 0.$$

From Theorem 4.3 we have that

$$(5.16a) \quad \frac{dH(n(t))}{dt} = - \sum_{s \in \mathcal{S}} \Gamma_s D\left(\frac{(\Lambda_r)_{r \in s}}{\Gamma_s} \left\| \frac{(n_r)_{r \in s}}{w_s}\right.\right) - \sum_{s \in \mathcal{S}} g_s(w_s) D\left(\frac{(n_r)_{r \in s}}{w_s} \left\| \frac{(\Lambda_r)_{r \in s}}{\Gamma_s}\right.\right)$$

$$(5.16b) \quad - \sum_{s \in \mathcal{S}} \left(\Gamma_s - g_s(w_s)\right) \left(\log \Gamma_s - \log g_s(w_s)\right).$$

It is clear that  $\frac{dH}{dt}$  is negative and combining the last two displays, (5.15) and (5.16), we have that

$$(5.17) \quad \frac{dH}{dt} = 0 \quad \text{iff} \quad g_s(w_s) = \Gamma_s(n) \quad \text{and} \quad \frac{n_r}{w_s} = \frac{\Lambda_r(n)}{\Gamma_s(n)}, \quad n_r > 0.$$

So  $H(n(t))$  is decreasing unless  $n(t)$  achieves the minimum of  $H(n)$ . Suppose that  $H(n(t))$  decreases to some value  $H'$  (which may or may not equal  $H^*$ ). There must be a sequence of times  $\{t_k\}_{k \in \mathbb{N}}$  such that  $\frac{dH(t_k)}{dt} \rightarrow 0$  as  $t_k \rightarrow \infty$ . Since  $w(t)$  is bounded from zero, it can not be that  $\Gamma_s(n(t_k)) \rightarrow 0$  as  $k \rightarrow \infty$ , so for this sequence of times it must be that

$$(5.18) \quad D\left(\frac{(n_r(t_k))_{r \in s}}{w_s} \left\| \frac{(\Lambda_r(t_k))_{r \in s}}{\Gamma_s(t_k)}\right.\right) \xrightarrow{k \rightarrow \infty} 0.$$

and

$$(5.19) \quad \left(\Gamma_s(t_k) - g_s(w_s(t_k))\right) \left(\log \Gamma_s(t_k) - \log g_s(w_s(t_k))\right) \xrightarrow{k \rightarrow \infty} 0.$$

Thus, by continuity of relative entropy at 0, any limit point,  $n^*, \Lambda^*, \Gamma^*$  of the sequences  $n(t_k), \Lambda(n(t_k)), \Gamma(n(t_k))$  must satisfy

$$\frac{\Lambda_r^*}{\Gamma_s^*} = \frac{n_r^*}{w_s} \quad \text{and} \quad g_s(w_s) = \Gamma_s(n)$$

for all  $r \in s$ ,  $s \in \mathcal{S}$ .

Thus the decreasing process  $H(t)$  must be decrease to the minimum it value  $H^*$ .

Finally, to consider the convergence of the processes  $n(t)$ ,  $w(t)$ ,  $\Lambda(t)$  and  $\Gamma(t)$ , we note that

$$H(t) = \tilde{H}(n(t), w(t), \Lambda(t)).$$

for the continuous convex function  $\tilde{H}$  given by (7.10).

$$\tilde{H}(n, w, \Lambda) = \sum_{s \in \mathcal{S}} \sum_{r \in s} n_r \log \frac{n_r}{w_s \Lambda_r} + \sum_{s \in \mathcal{S}} G_s(w_s).$$

It is clear that  $\tilde{H}$  has compact level sets and so must converge  $n(t)$ ,  $w(t)$  and  $\Lambda(t)$  must converge to the arguments minimizing  $\tilde{H}$  over  $\sum_{r \in s} n_r = w_s$ , for  $s \in \mathcal{S}$ ,  $\sum_{r \ni j} \Lambda_r \leq C_j$ ,  $j \in \mathcal{J}$ . As a consequence,  $\Gamma(t)$  must converge to the unique solution to the optimization (1.2).

So  $H(n(t))$  converges to  $H^*$ , we now show that  $\Gamma(n(t))$  converges to  $\Gamma^*$ . Observe

$$\begin{aligned} & H(n(t)) - H^* \\ &= \sum_{s \in \mathcal{S}} \sum_{r \in s} n_r(t) \log \left( \frac{n_r(t) \Gamma_s(n(t))}{w_s \Lambda_r(n(t))} \right) + \sum_{s \in \mathcal{S}} U_s(\Gamma_s^*) - \sum_{s \in \mathcal{S}} w_s \log(\Gamma_s(n(t))) \\ &\geq \sum_{s \in \mathcal{S}} U_s(\Gamma_s^*) - \sum_{s \in \mathcal{S}} w_s \log(\Gamma_s(n(t))) \\ &\geq 0. \end{aligned}$$

So since  $H(n(t)) - H^*$  converges to 0, and since  $\Gamma \mapsto \sum_s w_s \log \Gamma_s$  is continuous, it must be that, for  $s \in \mathcal{S}$ ,

$$(5.20) \quad \Gamma_s(n(t)) \xrightarrow[t \rightarrow \infty]{} \Gamma_s^*,$$

as required.  $\square$

**6. Acknowledgments.** The work of Jonatha Anselmi has been supported by grant MTM2010-17405 (Ministerio de Ciencia e Innovación, Spain) and grant SA-2012/00331 of the Department of Industry, Innovation, Trade and Tourism (Basque Government), which also sponsored a one-month visit of Neil Walton to the Basque Center for Applied Mathematics (BCAM). The work of Neil Walton is fund by the VENI research programme, which is financed by the Netherlands Organisation for Scientific Research (NWO).

## REFERENCES

- [1] J. Anselmi and G. Casale. Heavy-traffic revenue maximization in parallel multiclass queues. *Perform. Eval.*, 70(10):806–821, Oct. 2013.
- [2] J. Anselmi, B. D’Auria, and N. Walton. Closed queueing networks under congestion: Nonbottleneck independence and bottleneck convergence. *Math. Oper. Res.*, 38(3):469–491, Aug. 2013.
- [3] T. Bonald and L. Massoulié. Impact of fairness on internet performance. *Proc. of ACM Sigmetrics*, 29:82–91, 2001.
- [4] O. Bonaventure. Apple seems to also believe in multipath tcp, 2013.
- [5] Y. Cao, M. Xu, and X. Fu. Delay-based congestion control for multipath tcp. In *Proceedings of the 2012 20th IEEE International Conference on Network Protocols (ICNP)*, ICNP ’12, pages 1–10, Washington, DC, USA, 2012. IEEE Computer Society.



- [6] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. Tcp extensions for multipath operation with multiple addresses, January 2013. RFC 6824.
- [7] R. J. Gibbens, F. P. Kelly, and P. B. Key. Dynamic alternative routing. *Routing in communications networks*, pages 13–47, 1995.
- [8] T. J. Hacker, B. D. Athey, and B. Noble. The end-to-end performance effects of parallel tcp sockets on a lossy wide-area network. IPDPS '02, page 314, Washington, DC, USA, 2002. IEEE Computer Society.
- [9] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley. Multi-path tcp: A joint congestion control and routing scheme to exploit path diversity on the internet. *IEEE/ACM Trans. Networking*, 14(6):1261–1271, 2006.
- [10] M. Honda, Y. Nishida, L. Eggert, P. Sarolahti, and H. Tokuda. Multipath congestion control for shared bottleneck. In *7th International Workshop on Protocols for Future, Large-Scale and Diverse Network Transports (PFLDNeT)*, 2009.
- [11] D. R. Karger and M. Ruhl. Simple efficient load balancing algorithms for peer-to-peer systems. In *Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures*, pages 36–43. ACM, 2004.
- [12] F. P. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.
- [13] F. P. Kelly, A. K. Maulloo, and D. K. H. Tan. Rate control in communication networks: shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49:237–252, 1998.
- [14] F. P. Kelly and T. Voice. Stability of end-to-end algorithms for joint routing and rate control. *SIGCOMM Comput. Commun. Rev.*, 35(2):5–12, 2005.
- [15] F. P. Kelly and R. J. Williams. Fluid model for a network operating under a fair bandwidth-sharing policy. *Ann. Appl. Probab.*, 14:1055–1083, 2004.
- [16] P. Key, L. Massoulié, and D. Towsley. Path selection and multipath congestion control. In *In Proc. IEEE INFOCOM*, 2007.
- [17] R. Khalili, N. Gast, M. Popovic, and J.-Y. L. Boudec. Mptcp is not pareto-optimal: Performance issues and a possible solution. *IEEE/ACM Trans. Netw.*, 21(5):1651–1665, 2013.
- [18] L. Kleinrock. Time-shared systems: a theoretical treatment. *J. ACM*, 14(2):242–261, Apr. 1967.
- [19] S. Kullback and R. A. Leibler. On information and sufficiency. *Ann. Math. Statistics*, 22:79–86, 1951.
- [20] H. Kushner and P. Whiting. Convergence of proportional-fair sharing algorithms under general conditions. *Wireless Communications, IEEE Transactions on*, 3(4):1250–1259, 2004.
- [21] C. N. Laws. Resource pooling in queueing networks with dynamic routing. *Advances in Applied Probability*, 24(3):pp. 699–726, 1992.
- [22] T.-A. Le. Improving the performance of multipath congestion control over wireless networks. In *Advanced Technologies for Communications (ATC), 2013 International Conference on*, pages 60–65, Oct 2013.
- [23] X. Lin and N. Shroff. Utility maximization for communication networks with multipath routing. *Automatic Control, IEEE Transactions on*, 51(5):766–781, May 2006.
- [24] J. Little. A proof of the queueing formula  $L = \lambda W$ . *Operations Research*, pages 9:383–387, 1961.
- [25] S. H. Low and D. E. Lapsley. Optimization flow control, i: Basic algorithm and convergence. *IEEE/ACM TRANSACTIONS ON NETWORKING*, 7(6):861–874, 1999.
- [26] S. H. Low, L. L. Peterson, and L. Wang. Understanding tcp vegas: a duality model. *Journal of the ACM (JACM)*, 49(2):207–235, 2002.
- [27] J. Mo and J. Walrand. Fair end-to-end window-based congestion control. *IEEE/ACM Trans. Networking*, 8:556–567, 2000.
- [28] C. Paasch, G. Detal, F. Duchene, C. Raiciu, and O. Bonaventure. Exploring mobile/wifi handover with multipath tcp. In *Proceedings of the 2012 ACM SIGCOMM workshop on Cellular networks: operations, challenges, and future design*, CellNet '12, pages 31–36, New York, NY, USA, 2012. ACM.
- [29] Q. Peng, A. Walid, and S. H. Low. Multipath tcp algorithms: theory and design. SIGMETRICS '13, pages 305–316, New York, NY, USA, 2013. ACM.
- [30] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley. Improving datacenter performance and robustness with multipath tcp. SIGCOMM '11, pages 266–277, New York, NY, USA, 2011. ACM.
- [31] C. Raiciu, M. Handley, and D. Wischik. Coupled congestion control for multipath transport protocols. *RFC 6356*, <http://tools.ietf.org/html/rfc6356>, 2011.
- [32] C. Raiciu, C. Paasch, S. Barré, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Han-

- dley. How hard can it be? designing and implementing a deployable multipath tcp. In *USENIX Symposium of Networked Systems Design and Implementation (NSDI'12)*, San Jose (CA), 2012.
- [33] P. Schweitzer. Bottleneck determination in networks of queues. *Proc. ORSA/TIMS Special Interest Conf on Appl. Probab. - Comput. Sci., The Interface, Boca Raton, FLA*, pages 471–485, 1981.
- [34] P. J. Schweitzer. Approximate analysis of multiclass closed networks of queues. *Proceedings of the international conference on stochastic control and optimization*, 1979.
- [35] R. Srikant. *The Mathematics of Internet Congestion Control*. Birkhauser, 2004.
- [36] B. Ugaonkar, G. Pacifici, P. Shenoy, M. Spreitzer, and A. Tantawi. An analytical model for multi-tier internet services and its applications. *ACM SIGMETRICS*, pages 291–302, New York, NY, USA, 2005. ACM.
- [37] P. Viswanath, D. Tse, and R. Laroia. Opportunistic beamforming using dumb antennas. *Information Theory, IEEE Transactions on*, 48(6):1277–1294, 2002.
- [38] N. Walton. Concave switching in single and multihop networks. In *SIGMETRICS*. ACM, 2014.
- [39] N. S. Walton. Proportional fairness and its relationship with multi-class queueing networks. *Ann. Appl. Probab.*, 22(6):2301–2333, 2009.
- [40] W.-H. Wang, M. Palaniswami, and S. H. Low. Optimal flow control and routing in multi-path networks. *Perform. Eval.*, 52(2-3):119–132, Apr. 2003.
- [41] D. Wischik, M. Handley, and M. B. Braun. The resource pooling principle. *ACM SIGCOMM Computer Communication Review*, 38(5):47–52, 2008.
- [42] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. Design, implementation and evaluation of congestion control for multipath tcp. In *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, NSDI'11, pages 8–8, Berkeley, CA, USA, 2011. USENIX Association.
- [43] Y. Yi and M. Chiang. Stochastic network utility maximisation. *European Transactions on Telecommunications*, 19(4):421–442, 2008.
- [44] S.-Y. Yun and A. Proutiere. Distributed load balancing in heterogeneous systems. *CoRR*, abs/1403.4075, 2014.

**7. Proof of Additional Lemmas.** *Proof.* [Proof of Lemma 5.2] To prove that  $L(n)$  is convex, we use the dual formulation proven in Lemma 5.4. Let  $D(n, q)$  be the objective of the dual function (5.2). Observe  $D(n, q)$  is linear in  $n$ . For probabilities  $p = (p^0, p^1)$  with  $p^0 + p^1 = 1$  and for two vectors  $n^0, n^1 \in \mathbb{R}_+^{\mathcal{R}}$ , we define  $n^p = p^0 n^0 + p^1 n^1$ . We have

$$\begin{aligned} L(n^p) &= \max_{q \geq 0} D(n^p, q) = \max_{q \geq 0} \{p^0 D(n^0, q) + p^1 D(n^1, q)\} \\ &\leq p^0 \max_{q \geq 0} \{D(n^0, q)\} + p^1 \max_{q \geq 0} \{D(n^1, q)\} = p^0 L(n^0) + p^1 L(n^1). \end{aligned}$$

Thus, we see that  $L(n)$  is a convex function.

We consider the continuity of  $L(n)$ . Firstly, it is proven that  $n \mapsto \Lambda_r(n)$  is a continuous function for  $n_r > 0$ , see [15, Lemma A.3]. Thus each summand,  $r$ , in the above expression is continuous for  $n_r > 0$ . However, the case where  $n_r = 0$  remains. Since  $\Lambda_r(n) \leq C_{\max}$  and by bound (7.7) proven in Lemma 5.4,

$$(7.1) \quad n_r \log \frac{n_r}{C_{\max} w_s} \leq n_r \log \frac{n_r}{\Lambda_r(n) w_s} \leq n_r \log \frac{w_{\max}}{C_{\min} w_s}.$$

Thus,  $\lim_{n_r \rightarrow 0} n_r \log \frac{n_r}{\Lambda_r(n) w_s} = 0$ , which means that the summands of  $L(n)$  are also continuous at  $n_r = 0$ . Therefore,  $L(n)$  is continuous because it is the sum of continuous functions.

Finally, we note that the set of feasible values of  $n$  belongs to a compact set, so the set of optimal solutions is bounded and since  $L(n)$  is convex continuous the set of optimal solutions is also closed and convex.  $\square$

*Proof.* [Proof of Lemma 5.3] a) The function  $L(n)$  is the solution to the concave optimization

$$L(n) = \min - \sum_r n_r \log \left( \frac{\Lambda_r w_s}{n_r} \right) \text{ s.t. } \sum_{j \in r} \Lambda_r \leq C_j, j \in \mathcal{J} \text{ over } \Lambda_r \geq 0, r \in \mathcal{R}.$$

We calculate the dual of this optimization. After adding slack variables  $z_j$  and Lagrange multipliers  $q_j$ , the Lagrangian of this optimization is

$$(7.3) \quad - \sum_r n_r \log \left( \frac{\Lambda_r w_s}{n_r} \right) - \sum_j q_j \left( C_j - \sum_{r: j \in r} \Lambda_r - z_j \right).$$

We minimize such Lagrangian. As the slack variables are positive, it is clear that a finite solution  $q_j$  must be positive for each  $j \in \mathcal{J}$ , and by complementary slackness  $q_j z_j = 0$ . Differentiating with respect to  $\Lambda_r$ , we see that the Lagrangian of this optimization is minimized when

$$(7.4) \quad \frac{n_r}{\Lambda_r} = \sum_{j \in r} q_j.$$

Substituting this optimality condition back in to the Lagrangian (7.3), we see that the dual of the optimization is

$$\max_{q \geq 0} \sum_r n_r \log \left( \sum_{j \in r} q_j \right) - \sum_j q_j C_j + \sum_r n_r - \sum_s w_s \log w_s.$$

The primal problem, (7.2), satisfies Slater's condition and so is Strong Lagrangian. Hence, (5.2) holds.

b) We first place some bounds on the dual formulation proved above. The partial derivative of the objective, with respect to  $q_j$  for any  $j \in r$ , is

$$(7.5) \quad \frac{n_r}{\sum_{j \in r} q_j} - C_j.$$

For an optimal choice of  $q$  this quantity must be negative or equal to zero if  $q_j > 0$ . However, if we take  $\tilde{q}$  such that

$$(7.6) \quad \sum_{r: j \in r} \tilde{q}_j > \frac{w_{\max}}{C_{\min}}$$

then, (7.5) is negative. Thus, it must be that

$$(7.7) \quad \frac{n_r}{\Lambda_r(n)} = \sum_{j \in r} q_j \leq \frac{w_{\max}}{C_{\min}}$$

The equality holds by the optimality condition (7.6) holding. for the optimal  $q$ . The lowerbound is achieved by noting that  $\Lambda_r(n) \leq C_{\max}$  for all  $n$  and  $r$ . For each  $s$ , there must exists some route such that  $n_r \geq \frac{w_s}{|s|}$  and certainly  $\Gamma_s(n) \geq \Lambda_r(n)$  for each  $r \in s$ . Applying this to the above upper-bound derived, we have that

$$\frac{w_s}{|s| \Gamma_s(n)} \leq \frac{n_r}{\Lambda_r(n)} \leq \frac{w_{\max}}{C_{\min}}.$$

Rearranging gives,  $\Gamma_s(n) \geq \frac{w_{\max}|s|}{C_{\min}w_s} =: \Gamma_{\min}$ , as required.  $\square$

*Proof.* [Proof of Lemma 5.5] a) By the definition of  $G_s(w_s)$  (see (4.6)), we have

$$(7.8) \quad G_s(w_s) = \max_{\Gamma_s \geq 0} \{w_s \log \Gamma_s - U_s(\Gamma_s)\} = \max_{\gamma_s \in \mathbb{R}} \{w_s \gamma_s - U_s(e^{\gamma_s})\}$$

Thus,  $G_s(w_s)$  is the Legendre-Fenchel transform of the convex function  $U_s(e^{\gamma_s})$  and thus it is convex. The objective  $w_s \gamma_s - U_s(e^{\gamma_s})$  is concave provided the function  $U_s(e^{\gamma_s})$  is convex, which is equivalent to our assumption that  $\Gamma_s U'_s(\Gamma_s)$  is increasing. Differentiating this objective, we see that optimality is achieved by the  $\Gamma_s$  satisfying  $w_s = \Gamma_s U'_s(\Gamma_s)$ . In other words, the optimum for  $G_s(w_s)$  is achieved by  $\Gamma_s = g_s(w_s)$ . Next, we see that we can bound the derivative of  $G_s$  from the right as follows

$$\begin{aligned} & G_s(w_s + h) - G_s(w_s) \\ &= (w_s + h) \log g_s(w_s + h) - U_s(g_s(w_s + h)) - w_s \log(g_s(w_s)) + U_s(g_s(w_s)) \\ &\geq (w_s + h) \log(g_s(w_s)) - U_s(g_s(w_s)) - w_s \log(g_s(w_s)) + U_s(g_s(w_s)) = h \log(g_s(w_s)). \end{aligned}$$

By a symmetric argument we have  $h \log(g_s(w_s + h)) \geq G_s(w_s + h) - G_s(w_s)$ . Dividing by  $h$  and by the continuity of the function  $g_s$ , we have from that these bounds give

$$(7.9) \quad G'_s(w_s) = \log g_s(w_s).$$

This gives the derivative of  $G_s$ . Next, recall that  $G_s(w_s)$  is the Legendre-Fenchel transform of the convex function  $U_s(e^{\gamma_s})$ , see (7.8). Thus, Legendre-Fenchel duality holds and we can see that

$$U_s(e^{\gamma_s}) = \max_{w_s} \{w_s \gamma_s - G_s(-w_s)\}.$$

And so  $-U_s(\Gamma_s) = \min_{w_s} \{G_s(w_s) - w_s \log \Gamma_s\}$ . This proves part a).

b) We proceed in a similar manner to Lemma 5.1. We have

$$H(n) = \sum_{r \in \mathcal{R}} n_r \log n_r - \sum_{r \in \mathcal{R}} n_r \log \Lambda_r(n) - \sum_{s \in \mathcal{S}} w_s \log w_s + \sum_{s \in \mathcal{S}} G_s(w_s).$$

Differentiating the first two terms using standard arguments, differentiating the third term as in Lemma 5.1, and using (7.9), we get (5.11).  $\square$

*Proof.* [Proof of Lemma 5.6] We minimize  $H(n)$ . For this we consider the following sequence of equalities

$$\begin{aligned} & \min_{n \geq 0, w \geq 0} H(n) \quad \text{subject to} \quad \sum_{r \in s} n_r = w_s, \quad s \in \mathcal{S} \\ &= \min_{w \geq 0} \min_{\substack{n \geq 0: \\ \sum_{r \in s} n_r = w_s}} \min_{\substack{\Lambda, \Gamma \geq 0: \\ \sum_{r \in s} \Lambda_r = \Gamma_s \\ \sum_{r \ni j} \Lambda_r \leq C_j}} \sum_{r \in \mathcal{R}} n_r \log \frac{n_r}{w_s \Lambda_r} + \sum_{s \in \mathcal{S}} G_s(w_s) \\ &= \min_{\substack{\Lambda, \Gamma \geq 0: \\ \sum_{r \in s} \Lambda_r = \Gamma_s \\ \sum_{r \ni j} \Lambda_r \leq C_j}} \min_{w \geq 0} \min_{\substack{n \geq 0: \\ \sum_{r \in s} n_r = w_s}} \sum_{s \in \mathcal{S}} w_s \sum_{r \in s} \frac{n_r}{w_s} \log \frac{n_r \Gamma_s}{w_s \Lambda_r} + \sum_{s \in \mathcal{S}} \{G_s(w_s) - w_s \log \Gamma_s\} \\ &= \min_{\substack{\Lambda, \Gamma \geq 0: \\ \sum_{r \in s} \Lambda_r = \Gamma_s \\ \sum_{r \ni j} \Lambda_r \leq C_j}} \min_{w \geq 0} \sum_{s \in \mathcal{S}} \{G_s(w_s) - w_s \log \Gamma_s\} \end{aligned}$$

$$= \min_{\Lambda, \Gamma \geq 0} \sum_{s \in \mathcal{S}} U_s(\Gamma_s) \quad \text{subject to} \quad \sum_{r \ni j} \Lambda_r \leq C_j, \quad j \in \mathcal{J}, \quad \sum_{r \in s} \Lambda_r = \Gamma_s, \quad s \in \mathcal{S}.$$

In the first equality above, we introduce the optimization taken over  $\Lambda$ ; in the second equality, we reorder the minimization and add and subtract relevant terms; in the third, we minimize the above relative entropy term to zero; in the fourth we take the Fenchel transform of  $G_s(w_s)$  to obtain  $U_s$  and the required optimization.

The function

$$(7.10) \quad \tilde{H}(n, w, \Lambda) = \sum_{s \in \mathcal{S}} \sum_{r \in s} n_r \log \frac{n_r}{w_s \Lambda_r} + \sum_{s \in \mathcal{S}} G_s(w_s).$$

is convex over  $(\Lambda_r : r \in \mathcal{R})$  and  $(n_r : r \in \mathcal{R})$  and  $(w_s : s \in \mathcal{S})$  satisfying  $\sum_{r \in s} n_r = w_s$ , for  $s \in \mathcal{S}$ . Minimizing over  $\Lambda$  satisfying network capacity constraints  $\sum_{r \ni j} \Lambda_r \leq C_j$ ,  $j \in \mathcal{J}$ , we arrive at the proportionally fair optimization,  $\Lambda(n)$ , i.e.  $\tilde{H}(n) = H^*(n, w, \Lambda(n))$ . Minimizing over  $n_r$  (as in the third equality above) we obtain the condition

$$(7.11) \quad \frac{n_r}{w_s} = \frac{\Lambda_r}{\Gamma_s}$$

for each  $r \in \mathcal{R}$  (with  $r \in s$ ) such that  $n_r > 0$ . Further minimizing over  $w_s$  we obtain the condition

$$(7.12) \quad g_s(w_s) = \Gamma_s(n).$$

Together we see that the minimality condition for  $\tilde{H}$  and thus for  $H$  are (5.13) and (5.14).  $\square$